

## 周报

### 构建码表和索引体纹理

聚类完成以后，得到每个类的 ID( ClassID )以及其对应的体数据块 ID( VolumeBlockID )。首先通过聚类个数设置码表大小，然后统计每一个体数据块分别属于哪一个类，用一个数组记录统计结果，接着，对该结果进行筛选，统计数组元素值为 0 的元素个数（为 0 意味着没有体数据块属于当前类，说明当前类是无效类），用码表大小减去当前统计个数，调整码表大小，然后用调整后的数值开辟空间，计算码表元素的值，构建码表，同时使用上述得到的 ClassID、VolumeBlockID 以及每个体数据块在码表中的索引号 ActualID 对应关系构建索引体纹理。这种做法可以避免传统的向量量化方法中出现的无效聚类问题，节省空间，提高压缩率。

在测试数据上验证，其构建结果都是正确的。

### C++ & Matlab 混合编程

构建完码表和索引体纹理以后，开始测试实际的体数据，观察其压缩质量以及压缩率。在测试过程中，发现当前的计算特征值和特征向量的方法效率太低，（目前使用开源库 `eigen` 构建相关数据结构，编写特征值计算函数），在研究对比了几种开源库的计算效率后，采用 `Matlab` 的特征值计算函数。最简单也最直观的做法是将 c++ 程序中的拉普拉斯矩阵输出到文本文件，以该矩阵值作为 `matlab` 的输入，特征值和特征向量计算完成后再以文本的方式输出结果，然后在 c++ 程序中读入该数据。这种方法自然是可行的，只要将两端的接口和数据

结构衔接上即可，但是最大的问题在于这种做法使得程序被分离，执行过程会断开。因此，我将该方法列为最后的选择。

当前的方法为将 matlab 的函数编译为 dll 动态链接库，在 c++ 程序中以动态链接库的方式调用之。该方法效率和直接调用 matlab 差不多，也无需中断程序执行，缺点是混合编程相比之下，比较麻烦。摸索清楚 matlab 和 c++ 的混合编程方式后，完成了代码修改和测试。同时，由于数据结构的改变，不得不抛弃原有的 eigen 库，换成通用的使用基本 c++ 数据类型重构了整个谱聚类过程。

## 下周工作

将重新开始测试实际数据，给出压缩率、压缩质量等的报告，并探索几个参数的设置和压缩结果的关系。